

# Optimization of the Woodcock Particle Tracking Method Using Neural Network

Bingnan Zhang

University of Science and Technology of China

Hefei city, Anhui 230026

E-mail: zbn@mail.ustc.edu.cn

## Abstract

The acceptance rate in Woodcock tracking algorithm is generalized to an arbitrary position-dependent variable  $q(x)$ . A neural network is used to optimize  $q(x)$ , and the FOM value is used as the loss function. This idea comes from physics informed neural network(PINN), where a neural network is used to represent the solution of differential equations. Here the neural network  $q(x)$  should solve the functional equations that optimize FOM. For a 1d transmission problem with Gaussian absorption cross section, we observe a significant improvement of the FOM value compared to the constant  $q$  case and the original Woodcock method. Generalizations of the neural network Woodcock(NNW) method to 3d voxel models are waiting to be explored.

## 1 Introduction

Woodcock particle tracking method was first introduced in neutron transport by Woodcock et al[?]. Since its invention, many variants of Woodcock method have been implemented in Monte Carlo codes, for example Serpent [?], MORET[?], MONK[?], IMPC[?][?]. It has also been applied in radioactive transfer[?], medical physics [?][?], computer graphics[?][?][?], and other areas. In Monte Carlo simulation, collision distance is usually calculated by inverting the integral of cross section along the particle path. However, this is very difficult when the media is in-homogeneous. Woodcock tracking takes a fictitious cross section  $\Sigma_s >= \max(\Sigma(x))$  as the majorant to sample collision distance, then the distance is accepted with probability  $\Sigma(x)/\Sigma_s$ . When the distance is not accepted, we say there is a null collision. Since the sampling cross section  $\Sigma_s$  is a constant, it is straightforward to sample the collision distance  $d = -\ln\zeta/\Sigma_s$ , where  $\zeta$  follows a uniform distribution on  $[0, 1]$ . However, in problems where there are regions with significantly bigger cross sections than other places, for example the control rods in nuclear reactors, choosing the majorant as the sampling cross section can be cumbersome, as it will generate a lot of unnecessary null collisions. In [?][?][?], a biased version of Woodcock tracking is proposed, where the sampling cross section is an arbitrary constant, and the acceptance rate is also a variable. The particle weight is adjusted after every collision to compensate for these changes. This biased version of Woodcock tracking can significantly improve the FOM value, which is defined as  $\frac{1}{\text{relative\_error}^2 \cdot \text{run\_time}}$  to measure the efficiency, if the parameters are

chosen properly. The acceptance rate can also be a position-dependent function rather than a constant[?]. This in principle can further improve the efficiency because there are more degree of freedoms to tune. However, since the problem becomes more complicated, optimization also requires more effort. In this paper, a neural network inspired by PINN[?][?] is used to optimize the sampling cross section and the acceptance rate, and they are tested in a 1d transmission problem.

Section ?? is a general introduction of Woodcock tracking and biased Woodcock tracking. Section ?? states the transmission problem and analytically derives FOM. Section ?? and ?? describe the numerical optimization and simulation results. Section ?? includes a summary and future research directions.

## 2 The Woodcock tracking formalism

As described in section I, the Woodcock tracking method is characterized by a majorant cross section  $\Sigma_s$ , and the distance is sampled from distribution

$$f(x) = \Sigma_s e^{-\Sigma_s x} \quad (1)$$

which is accepted with probability  $\Sigma(x)/\Sigma_s$ . This can be described by the following code:

```
w=1;
while !collided:
     $\zeta_1 = rand();$ 
     $s = -ln(\zeta_1)/\Sigma_s;$ 
     $\mathbf{x}=\mathbf{x}+\Omega s;$ 
     $\zeta_2 = rand();$ 
    if  $\zeta_2 < \Sigma(\mathbf{x})/\Sigma_s:$ 
        collided=true;
        handle real collision;
    else:
        null collision,do nothing;
    end
end
```

A mathematical proof of the correctness can be found in [?]. In biased Woodcock tracking, however, the acceptance rate becomes a variable, which is denoted as  $q$ . The particle weight is adjusted at every step to compensate for this change:

```
w=1;
while !collided:
     $\zeta_1 = rand();$ 
     $s = -ln(\zeta_1)/\Sigma_s;$ 
     $\mathbf{x}=\mathbf{x}+\Omega s;$ 
     $\zeta_2 = rand();$ 
    if  $\zeta_2 < q:$ 
        collided=true;
```

```

 $w = w^{\frac{\Sigma(x)}{\Sigma_s q}};$ 
    handle real collision;
else:
     $w = w^{\frac{1-\Sigma(x)/\Sigma_s}{1-q}};$ 
end
end

```

The unbiasedness of such a modification can be proved via the probability generating function(PGF) of in-homogeneous Poisson process(IHPP)[?]. When  $\Sigma_s < \Sigma(x)$ , the particle weight turns negative in a null collision event[?]. These negative weight particles can be understood as fictitious anti-particles that will cancel the over-weight of other particles. Various authors have proposed different forms of  $q$ [?][?], however, little effort has been devoted to its theoretical or numerical optimization. When  $q(x)$  is a position-independent constant, a plot of FOM with respect to  $\Sigma_s$  and  $q$  has been made in [?], and the optimal FOM is about 7 times bigger than Carter's algorithm[?]. In this paper, we focus on the position-dependent  $q(x)$ , and tries to optimize it using neural network. This should inspire people to generalize it to 3d voxel models.

### 3 The transmission problem

The problem we consider in this paper is a 1d transmission problem with narrow Gaussian absorption cross section (figure ??). The absorption peak lies at  $x = 1.23$ .

$$\Sigma(x) = 0.8e^{-\frac{(x-1.23)^2}{0.1^2}} + 0.1 \quad (2)$$

Particles fly from  $x = 0$  to the right and we record the number of particles that are not ab-

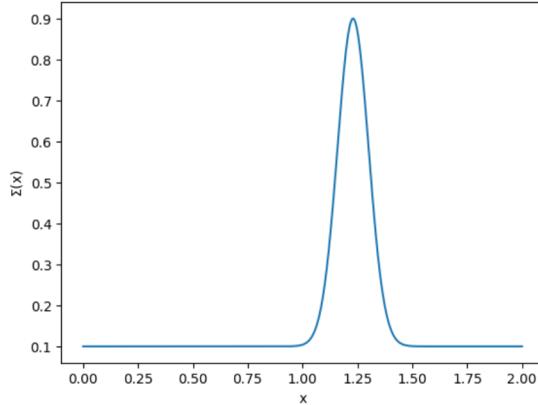


Figure 1: The narrow Gaussian cross section  $\Sigma(x)$

sorbed within the  $0 < x < 2$  region. The algorithm is summarized below:

```

w=1;
while !collided:
     $\zeta_1 = rand();$ 
     $s = -ln(\zeta_1)/\Sigma_s;$ 

```

```

x=x+s;
if  $x > L$ :
    record transmission event;
    kill the particle;
else:
     $\zeta_2 = \text{rand}();$ 
    if  $\zeta_2 < q(x)$ :
        collided=true;
        kill the particle;
    else:
         $w = w \frac{1 - \Sigma(x) / \Sigma_s}{1 - q(x)};$ 
    end
end
end

```

Following [?], we derive the relative error of the particle weight  $w$ . The expectation value can be expanded with respect to null collision events.

$$E[w] = \sum_{k=0}^{\infty} \prod_{i=1}^k \int dx_i E[w | X_1 = x_1, X_2 = x_2, \dots, X_k = x_k] \\ \prod_{i=1}^k (1 - q(x_i)) p_{X|k}(x_1, x_2, \dots, x_k) p_k \quad (3)$$

where  $p_k$  is the probability that the particle undergoes  $k$  collisions before reaching  $x = 2$ .  $p_{X|k}(x_1, x_2, \dots, x_k)$  represents the probability that such collisions happen at  $x_1, x_2, \dots, x_k$ . The integral doesn't change if we shuffle the order of  $x_1, x_2, \dots, x_k$ , so we ignore the constraint that  $x_1 < x_2 < \dots < x_k$ . Since  $\Sigma_s$  is constant, the probability density should be uniform[?], so

$$p_{X|k}(x_1, x_2, \dots, x_k) = 1/L^k \quad (4)$$

where  $L = 2$ .  $\prod_{i=1}^k (1 - q(x_i))$  makes sure that all collisions are null so that the particle is not killed in the middle.  $E[w | X_1 = x_1, X_2 = x_2, \dots, X_k = x_k]$  is the particle weight in such a transmission event. Under the biased Woodcock formalism

$$E[w | X_1 = x_1, X_2 = x_2, \dots, X_k = x_k] = \prod_{i=1}^k \frac{1 - \Sigma(x_i) / \Sigma_s}{1 - q(x_i)} \quad (5)$$

Put these into equation (??), we get

$$E[w] = \sum_{k=0}^{\infty} \left(1 - \frac{\bar{\Sigma}}{\Sigma_s}\right)^k p_k \quad (6)$$

where  $\bar{\Sigma}$  is the mean of  $\Sigma(x)$  on the line. We can also calculate the second moment in the same

way

$$\begin{aligned}
E[w^2] &= \sum_{k=0}^{\infty} \prod_{i=1}^k \int dx_i E[w^2 | X_1 = x_1, X_2 = x_2, \dots, X_k = x_k] \\
&\quad \prod_{i=1}^k (1 - q(x_i)) p_{X|k}(x_1, x_2, \dots, x_k) p_k \\
&= \sum_{k=0}^{\infty} \prod_{i=1}^k \int dx_i \prod_{i=1}^k \frac{(1 - \Sigma(x_i)/\Sigma_s)^2}{1 - q(x_i)} \frac{1}{L^k} p_k \\
&= \sum_{k=0}^{\infty} \left( \frac{1}{L} \int dx \frac{(1 - \Sigma(x)/\Sigma_s)^2}{1 - q(x)} \right)^k p_k
\end{aligned} \tag{7}$$

The number of collisions  $k$  follows Poission distribution with PGF

$$G(z) = \sum_{k=0}^{\infty} z^k p_k = \exp((z-1)L\Sigma_s) \tag{8}$$

So the first and second moment of  $w$  can be calculated

$$E[w] = \exp\left(-\int_0^L dx \Sigma(x)\right) \tag{9}$$

$$E[w^2] = \exp\left\{\int_0^L dx \frac{q(x)\Sigma_s - 2\Sigma(x) + \Sigma(x)^2/\Sigma_s}{1 - q(x)}\right\} \tag{10}$$

The squared relative error is

$$\begin{aligned}
r^2 &= \frac{1}{N} \frac{E[w^2] - E[w]^2}{E[w]^2} \\
&= \frac{1}{N} \left( \exp\left(\int_0^L \frac{q(x)(\Sigma_s - 2\Sigma(x)) + \Sigma(x)^2/\Sigma_s}{1 - q(x)}\right) - 1 \right)
\end{aligned} \tag{11}$$

where  $N$  is the number of particles.

The run time is roughly proportional to the number of cross section evaluations [?][?], which is also the number of null collisions. We have

$$T = N \int_0^L dx \Sigma_s (1 - q(x)) e^{-\int_0^x \Sigma_s q(y) dy} \tag{12}$$

the exponential  $e^{-\int_0^x \Sigma_s q(y) dy}$  is the probability that a particle is not absorbed on  $[0, x]$ , and  $\Sigma_s (1 - q(x)) dx$  is the number of null collisions such a particle contributes on  $dx$ . Note that this expression for  $T$  is different from the one used in [?]. The derivation of  $T$  in [?] is attached in appendix ??.

Finally, the FOM is defined as

$$\begin{aligned}
FOM &= \frac{1}{Tr^2} \\
&= \frac{1}{\exp\left(\int_0^L dx \frac{q(x)(\Sigma_s - 2\Sigma(x)) + \Sigma(x)^2/\Sigma_s}{1 - q(x)}\right) - 1} \\
&\quad \cdot \frac{1}{\int_0^L dx \Sigma_s (1 - q(x)) e^{-\int_0^x \Sigma_s q(y) dy}}
\end{aligned} \tag{13}$$

which measures the efficiency of the algorithm. One interesting limit is  $q = 0, \Sigma_s \gg \Sigma(x)$ . In this limit, null collisions are sampled with high frequency and there are no real collisions. FOM reduces to

$$\begin{aligned} FOM &\approx \frac{1}{\exp(\int_0^L \Sigma(x)^2 dx / \Sigma_s) - 1} \frac{1}{\Sigma_s L} \\ &\approx \frac{1}{L \int_0^L \Sigma(x)^2 dx} \end{aligned} \quad (14)$$

This limit is not yet covered by the neural network version of  $q(x)$ . As described in section ??,  $\Sigma_s$  is obtained from  $q(-1)$  that is bounded by the activation function  $\tanh$ . One may change the output layer to cover this limit.

## 4 Optimization

Solving for the optimal  $q(x)$  that maximize equation ?? is an extremely complicated functional optimization problem, and the author doesn't think an analytical solution can be obtained. Instead, numerical methods should be applied. The widely used machine learning packages provide us perfect tools for solving such optimization problems. This is the basic idea behind physics informed neural network(PINN)[?][?]. In this paper, we use a simple FFN network with  $\tanh$  activation, whose output range is  $[-1, 1]$ . The input data is position  $x$ , and the output when  $0 < x < 2$  is interpreted as  $2q(x) - 1$ . The output at  $x = -1$  is interpreted as  $\Sigma_s / \Sigma_m - 1$ , where  $\Sigma_m$  is the maximum value of  $\Sigma(x)$ . Obviously, this choice is not unique.

We discretize the space  $[0, 2]$  and calculate the integral in equation ?? with the Pytorch function trapz. An Adam optimizer is used to maximize FOM. This process is fast because the FFN network used in this paper is small. The change of FOM with training epoch is shown in figure ??.

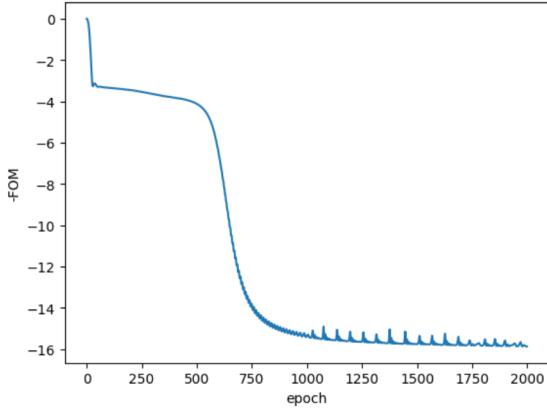


Figure 2: The change of FOM with training epochs.

The optimized  $q(x)$  is plotted in figure ?? . Remember in the original version of Woodcock tracking,  $q(x)$  is proportional to  $\Sigma(x)$  shown in figure ?? . Here the optimized  $q(x)$  looks exactly the opposite.  $\Sigma_s = 0.099$  is obtained from  $q(-1)$ .

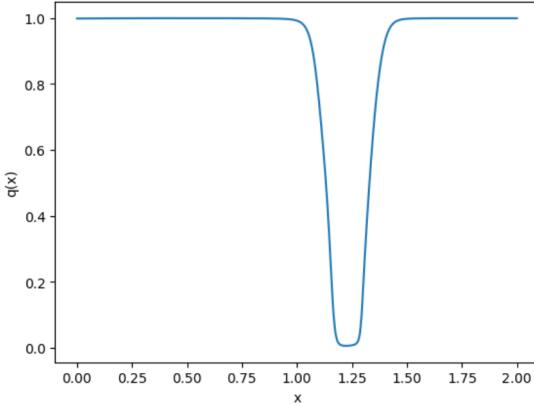


Figure 3:  $q(x)$  after optimization.

## 5 Simulation results

Simulations are conducted using several different Woodcock-type tracking methods. The neural network  $q(x)$  obtained in section ?? is used to run the neural network Woodcock(NNW) simulation. The relative error  $r$  of transmission rate and the number of cross section evaluations  $T$  are recorded. The change of  $Nr^2$  and  $T/N$  with particle number  $N$ , where  $N$  is introduced to normalize the result, are plotted in figure ??,???. The theoretical predictions are calculated from equation ??,??. The relative error and cross section evaluation number converge to the theoretical values when  $N > 10^7$ , and the correctness of the derivations in section ?? are proven.

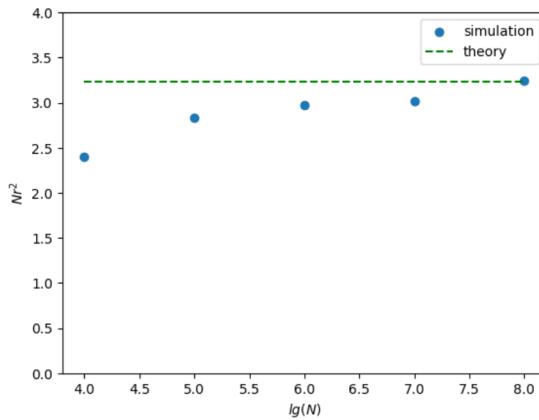


Figure 4: The convergence of  $Nr^2$  towards theoretical prediction.

A summary of  $Nr^2$ ,  $T/N$ ,  $FOM$  for different Woodcock-type tracking methods is made in table ??.

The first column in table ?? represents the original Woodcock, the second column is the biased Woodcock using constant  $q$ ,  $\Sigma_s$ , the third column is the biased Woodcock using the optimized neural network  $q(x)$ . The constant biased Woodcock method is also used in [?], but the parameters are not optimized. Here a grid search is conducted to find the optimal  $q$ ,  $\Sigma_s$ , and the second column records the best result.

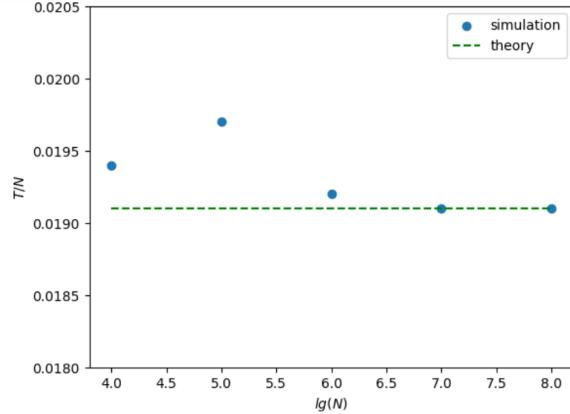


Figure 5: The convergence of  $T/N$  towards theoretical prediction.

Table 1: Comparison of the original Woodcock, constant  $q$  biased Woodcock(CBW), and neural network Woodcock(NNW) methods using narrow Gaussian cross section.

	Woodcock	CBW	NNW
$Nr^2$	0.104	0.000727	2.94
T/N	1.27	89.8	0.0198
FOM	1.95	3.91	17.2

The constant biased Woodcock is about twice as efficient as the original Woodcock, while using function  $q(x)$  and neural network further improves the efficiency by 340%. Note that the efficiency improvement comes from  $T$  while the relative error gets bigger.

In table ??, more simulations are done using several different cross sections and the FOM values are recorded. The optimal constants are already used in constant biased Woodcock. The neural network Woodcock beats constant biased Woodcock in all cases, and the FOM improvement gets bigger when the cross section peaks become sharper.

Sometimes the constant  $q$  biased Woodcock method gives the largest FOM in the  $q = 0, \Sigma_s >> 1$  limit. Obviously, this limit is not covered by the neural network because the range of  $\Sigma_s$  is constrained by the  $\tanh$  activation function. Using other functions that covers the whole  $\Sigma_s > 0$  region in the output layer should in principle further improve the efficiency.

## 6 Conclusion

In this paper, we optimize the acceptance probability  $q(x)$  and sampling cross section  $\Sigma_s$  in biased Woodcock tracking method. A neural network is used to represent these variables, and the FOM value is used as the loss function. In a 1d transmission problem with narrow Gaussian absorption cross section, we observe a significant improvement of the FOM value compared to the constant  $q$  biased Woodcock and also the original version of Woodcock tracking. This

Table 2: Comparison of constant q biased Woodcock(CBW) and neural network Woodcock(NNW) using different cross sections. FOM values are recorded.

$\Sigma(x)$	CBW	NNW	Improvement
$0.8e^{-\frac{(x-1.23)^2}{0.1^2}} + 0.1$	3.91	17.2	340%
$0.8e^{-\frac{(x-1.23)^2}{1^2}} + 0.1$	1.40	1.91	36.4%
$\sin(3x) + 1$	0.151	0.408	170%
$e^{-x}$	1.24	1.84	48.4%

improvement mainly comes from the significant reduction of cross section evaluations. The correctness of the theoretical derivations using IHPP process is supported by the simulation result. Several other cross sections are also used to run the simulation and the efficiency improvement is demonstrated. One may want to change the output layer to cover the  $\Sigma_s \gg \Sigma(x)$  limit and other optimization methods without neural network can also be explored. In order to make this method more applicable in industrial simulations, future research can also be done on generalizing this method to 3d voxel models. In this case, it might be difficult to obtain the analytical expression of FOM, and a self-adaptive method could be more realistic.

## A

A different type of computational cost estimation formula is given in [?]. This appendix includes their derivation. The computational cost mainly comes from the effort to generate a real collision, so we estimate the number of null collisions generated before a real collision is accepted. Let C be the number of null collisions until a real collision is generated. The computational cost is estimated by the expectation of C as follows:

$$E[C] = \int_0^\infty \left( \int_0^x \Sigma_s dy \right) q(x) \Sigma_s \exp \left( - \int_0^x q(y) \Sigma_s dy \right) dx \quad (15)$$

This works when the space is infinite. When the space is finite, however, the particle can escape the region before a real collision happens. A modification of the above equation gives

$$\begin{aligned} E[C_L] &= \int_0^L \left( \int_0^x \Sigma_s dy \right) q(x) \Sigma_s \exp \left( - \int_0^x q(y) \Sigma_s dy \right) dx \\ &\quad + \int_0^L \Sigma_s dx \cdot \exp \left( - \int_0^L q(x) \Sigma_s dx \right) \end{aligned} \quad (16)$$

When  $\Sigma_s$  is a constant, the above equation reduces to

$$\begin{aligned} E[C_L] &= \int_0^L \Sigma_s^2 x q(x) \exp \left( - \int_0^x q(y) \Sigma_s dy \right) dx \\ &\quad + \Sigma_s L \cdot \exp \left( - \int_0^L q(x) \Sigma_s dx \right) \end{aligned} \quad (17)$$

where  $L$  is the length of the segment.

## References

- [1] B.Molnar,D.Legrady, *Variance analysis of Woodcock type tracking*,International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering(2017)
- [2] B.Molnar, G.Tolnai, and D.Legrady, *Variance reduction and optimization strategies in a biased Woodcock particle tracking framework*, Nuclear Science and Engineering, 190, 56–72(2018).
- [3] D. Legrady,B. Molnar,M. Klausz and T. Major, *Woodcock tracking with arbitrary sampling cross section using negative weights*,Annals of Nuclear Energy,102,116-123(2017).
- [4] H.Belangera, D.Mancusib, A.Zoiac,*Review of Monte Carlo methods for particle transport in continuously-varying media*, Eur. Phys. J. Plus 135:877(2020)
- [5] E. Woodcock et al., *Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry*, Proc. Conf. Appl. Comput. Methods Reactor Probl., 557, 2 (1965);
- [6] L.Carter, E.Cashwell, and W.Taylor, *Monte Carlo sampling with continuously varying cross section along flight paths*, Nuclear Science and Engineering, 48,4, 403–411 (1972).
- [7] V.S.Antyufeev, *Mathematical verification of the Monte Carlo maximum cross-section technique*, Monte Carlo Methods Appl., 21, 4, 275 (2015)
- [8] P.V.Mieghem,*Performance analysis of complex networks and systems*,Cambridge University Press(2009).
- [9] J.Leppanen, *Performance of Woodcock delta-tracking in lattice physics applications using the Serpent Monte Carlo reactor physics burnup calculation code*, Ann. Nucl. Energy, 37, 5, 715 (2010);
- [10] B.Forestier et al.,*Criticality calculations on Pebble Bed HTR-PROTEUS configuration as a validation for the psedo-scattering tracking method implemented in the MORET 5 Monte Carlo Code*, Int. Conf. Phys. Reactors,Interlaken, Switzerland, September 14–19, 2008.
- [11] S.D.Richards et al., *MONK and MCBEND: current status and recent developments*, Ann. Nucl. Energy, 82, 63 (2015);

- [12] P.Fang,X.Wu,Y.Yang,H.Wang,L.Yang,Y.Guo and H.Lai *Development and preliminary verification of a Monte Carlo neutron transport code IMPC-Neutron*,Annals of Nuclear Energy,175,109221(2022)
- [13] P.Fang,Y.Yang,H.Wang,L.Yang,Y.Guo,H.Lai, and X.Wu,*Implementation and optimization of WDT algorithm on IMPC-Neutron and comparison of efficiency of various algorithms*,Annals of Nuclear Energy,187,109786(2023)
- [14] J.Novak, A.Selle, and W.Jarosz, *Residual ratio tracking for estimating attenuation in participating media*, ACM Transactions on Graphics (TOG), 33, 6, 179 (2014).
- [15] M.Raab, D.Seibert, and A.Keller, *Unbiased global illumination with participating media*,Monte Carlo and Quasi-Monte Carlo Methods 2006, Springer, pp. 591–605 (2008).
- [16] R.Sosan,M.M.Movania and S.Siddiqui,*Perceptual analysis of distance sampling and transmittance estimation techniques in biomedical volume visualization*,Turkish J. Electr. Eng. Comput. Sci.,30,2109-2123(2022)
- [17] V.Eymet,D.Poitou,M.Galtier,M.Elhafi,G. Terree, and R.Fournier, *Null-collision meshless Monte-Carlo-Application to the validation of fast radiative transfer solvers embedded in combustion simulators*, Journal of Quantitative Spectroscopy and Radiative Transfer, 129, 145–157 (2013).
- [18] A. BADAL and A. BADANO, *Monte Carlo simulation of X-ray imaging using a graphics processing unit*, 2009 IEEE Nuclear Science Symposium Conference Record (NSS/MIC), IEEE (2009), pp. 4081–4084.
- [19] Z.Liu, C.Zheng, N.Zhao, Y.Huang, J.Chen, Y.Yang, *A GPU-accelerated Monte Carlo dose computation engine for small animal radiotherapy*,Medical Physics,50,5238-5247(2023)
- [20] L.WIG.Morgan and D.Kotlyer, *Weighted Delta Tracking for Monte Carlo Particle Transport*, Ann.Nucl.Energy,85,1184(2015)
- [21] M.Galtier et al.,*Integral Formulation of Null Collision Monte Carlo Algorithms*,J. Quant. Spectrosc.Radiat. Transf., 125, 57 (2013);
- [22] L. SZIRMAY-KALOS et al., *Unbiased Light Transport Estimators for Inhomogeneous Participating Media*,MOLNAR et al. Comput. Graphics Forum, 36, 2, 9 Wiley Online Library (2017)
- [23] H.Hu,L.Qi and X.Chao,*Physics-informed Neural Networks (PINN) for computational solid mechanics: Numerical frameworks and applications*,Thin-Walled Structures,205,112495(2024)
- [24] G.E.Karniadakis,Y.Kevrekidis,L.Lu,P.Perdikaris,*Physics-informed machines learning*. Nature Reviews Physics(2021)